DM545

Linear and Integer Programming

Lecture 11
Network Flows

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. (Minimum Cost) Network Flows

2. Duality in Network Flow Problems

3. Assignment and Transportation

Outline

1. (Minimum Cost) Network Flows

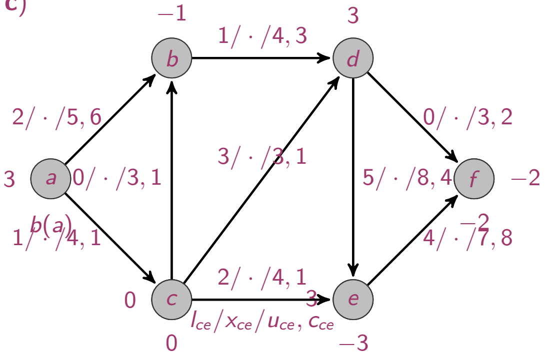
2. Duality in Network Flow Problems

3. Assignment and Transportation

Terminology

Network:
- directed graph $D = (V, A)$
- arc, directed link, from tail to head
- lower bound $l_{ij} > 0$, $\forall ij \in A$, capacity $u_{ij} \geq l_{ij}$, $\forall ij \in A$
- cost c_{ij}, linear variation (if $ij \notin A$ then $l_{ij} = u_{ij} = 0$, $c_{ij} = 0$)
- balance vector $b(i)$, $b(i) > 0$ supply node (source), $b(i) < 0$ demand node (sink, tank), $b(i) = 0$ transhipment node (assumption $\sum_i b(i) = 0$)

$N = (V, A, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mathbf{c})$



Network Flows

Flow $\mathbf{x} : A \to \mathbb{R}$

balance vector of $\mathbf{x}$: $b_{\mathbf{x}}(v) = \sum_{vu \in A} x_{vu} - \sum_{wv \in A} x_{wv}, \forall v \in V$

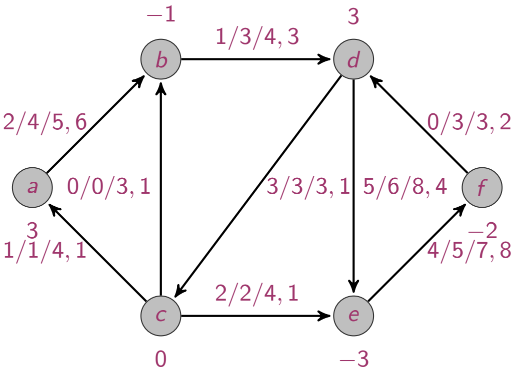
$$b_{\mathbf{x}}(v) \begin{cases} > 0 & \text{source} \\ < 0 & \text{sink/target/tank} \\ = 0 & \text{balanced} \end{cases}$$

(generalizes the concept of path with $b_{\mathbf{x}}(v) = \{0, 1, -1\}$)

feasible $\quad l_{ij} \le x_{ij} \le u_{ij}$, $b_{\mathbf{x}}(i) = b(i)$
cost $\quad \mathbf{c}^T\mathbf{x} = \sum_{ij \in A} c_{ij} x_{ij}$ (varies linearly with $\mathbf{x}$)

If iji is a 2-cycle and all $l_{ij} = 0$, then at least one of x_{ij} and x_{ji} is zero.

Example



Feasible flow of cost 109

Minimum Cost Network Flows

Find cheapest flow through a network in order to satisfy demands at certain nodes from available supplier nodes.

Variables:

$$x_{ij} \in \mathbb{R}_0^+$$

Objective:

$$\min \sum_{ij \in A} c_{ij} x_{ij}$$

Constraints: mass balance + flow bounds

$$\sum_{j:ij \in A} x_{ij} - \sum_{j:ji \in A} x_{ji} = b(i) \quad \forall i \in V$$

$$l_{ij} \leq x_{ij} \leq u_{ij}$$

$$\begin{aligned} \min\ & \mathbf{c}^T \mathbf{x} \\ & N\mathbf{x} = \mathbf{b} \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned}$$

N node arc incidence matrix

(assumption: all values are integer, we can multiply if rational)

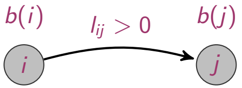
$$
\begin{array}{r:cccccccc}
 & x_{e_1} & x_{e_2} & \cdots & x_{ij} & \cdots & x_{e_m} & & \\
 & c_{e_1} & c_{e_2} & \cdots & c_{ij} & \cdots & c_{e_m} & & \\
\hdashline
1 & -1 & . & \cdots & . & \cdots & . & = & b_1 \\
2 & . & . & \cdots & . & \cdots & . & = & b_2 \\
\vdots & \vdots & \ddots & & & & & = & \vdots \\
i & 1 & . & \cdots & -1 & \cdots & . & = & b_i \\
\vdots & \vdots & \ddots & & & & & = & \vdots \\
j & . & . & \cdots & 1 & \cdots & . & = & b_j \\
\vdots & \vdots & \ddots & & & & & = & \vdots \\
n & . & . & \cdots & . & \cdots & . & = & b_n \\
\hdashline
e_1 & 1 & & & & & & \leq & u_1 \\
e_2 & & 1 & & & & & \leq & u_2 \\
\vdots & \vdots & \ddots & & & & & \leq & \vdots \\
(i,j) & & & & 1 & & & \leq & u_{ij} \\
\vdots & \vdots & \ddots & & & & & \leq & \vdots \\
e_m & & & & & & 1 & \leq & u_m
\end{array}
$$

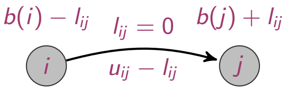
Reductions/Transformations

Lower bounds

Let $N = (V, A, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mathbf{c})$

$$
\begin{aligned}
N' &= (V, A, \mathbf{l}', \mathbf{u}', \mathbf{b}', \mathbf{c}) \\
b'(i) &= b(i) - l_{ij} \\
b'(j) &= b(j) + l_{ij} \\
u'_{ij} &= u_{ij} - l_{ij} \\
l'_{ij} &= 0
\end{aligned}
$$





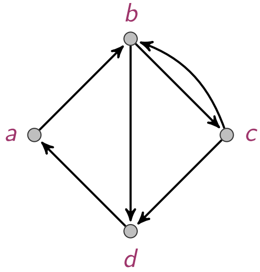
$$\mathbf{c}^T\mathbf{x}$$

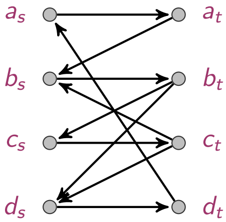
$$\mathbf{c}^T\mathbf{x}' + \sum_{ij \in A} c_{ij} l_{ij}$$

Undirected arcs

Vertex splitting

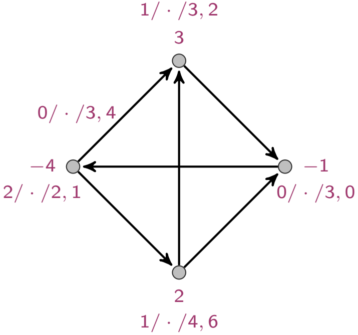
If there are bounds and costs of flow passing through vertices where $b(v) = 0$ (used to ensure that a node is visited):
$N = (V, A, \mathbf{l}, \mathbf{u}, \mathbf{c}, \mathbf{l}^*, \mathbf{u}^*, \mathbf{c}^*)$

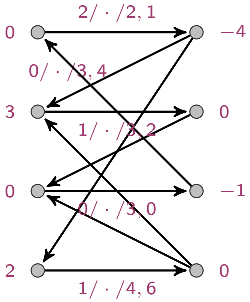




From D to D_{ST} as follows:

$$\begin{aligned} \forall v \in V \quad & \rightsquigarrow v_s, v_t \in V(D_{ST}) \text{ and } v_s v_t \in A(D_{ST}) \\ \forall xy \in A(D) & \rightsquigarrow x_t y_s \in A(D_{ST}) \end{aligned}$$



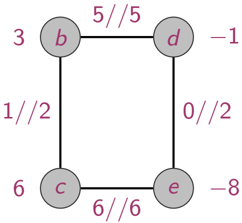


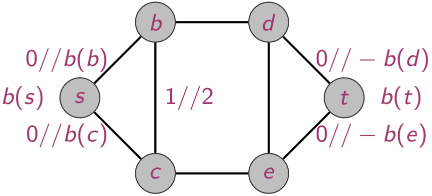
$$\forall v \in V \text{ and } v_s v_t \in A_{ST} \rightsquigarrow h'(v_s v_t) = h^*(v), \quad h^* \in \{l^*, u^*, c^*\}$$
$$\forall xy \in A \text{ and } x_t y_s \in A_{ST} \rightsquigarrow h'(x_t y_s) = h(x, y), \; h \in \{l, u, c\}$$

If $b(v) = 0$, then $b'(v_s) = b'(v_t) = 0$
If $b(v) < 0$, then $b'(v_s) = 0$ and $b'(v_t) = b(v)$
If $b(v) > 0$, then $b'(v_s) = b(v)$ and $b'(v_t) = 0$

(s,t)-flow:

$$b_{\mathbf{x}}(v) = \begin{cases} k & \textit{if } v = s \\ -k & \textit{if } v = t \\ 0 & \textit{otherwise} \end{cases}, \qquad |\mathbf{x}| = |b_{\mathbf{x}}(s)|$$





$$b(s) = \sum_{v:b(v)>0} b(v) = M$$
$$b(t) = \sum_{v:b(v)<0} b(v) = -M$$

$\exists$ feasible flow in N $\iff$ $\exists$ (s,t)-flow in N_{st} with $|x| = M$ $\iff$ max flow in N_{st} is M

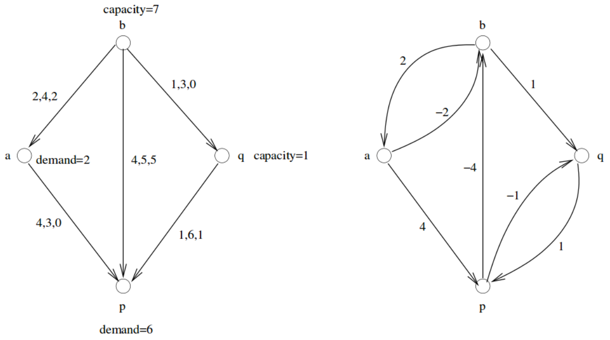
Residual Network

Residual Network $N(\mathbf{x})$: given that a flow $\mathbf{x}$ already exists, how much flow excess can be moved in G?

Replace arc $ij \in N$ with arcs:

	residual capacity	cost
ij :	$r_{ij} = u_{ij} - x_{ij}$	c_{ij}
ji :	$r_{ji} = x_{ij}$	$-c_{ij}$

$(N, \mathbf{c}, \mathbf{u}, \mathbf{x})$ $\qquad$ $(N(\mathbf{x}), \mathbf{c}')$



Special cases

Shortest path problem path of minimum cost from s to t with costs $\lesseqgtr 0$

$b(s) = 1, b(t) = -1, b(i) = 0$

if to any other node? $b(s) = (n-1), b(i) = 1, u_{ij} = n-1$

Max flow problem incur no cost but restricted by bounds

steady state flow from s to t

$b(i) = 0 \ \forall i \in V, \qquad c_{ij} = 0 \ \forall ij \in A \qquad ts \in A$

$c_{ts} = -1, \qquad u_{ts} = \infty$

Assignment problem min weighted bipartite matching,

$|V_1| = |V_2|, A \subseteq V_1 \times V_2$

c_{ij}

$b(i) = 1 \ \forall i \in V_1 \qquad b(i) = -1 \ \forall i \in V_2 \qquad u_{ij} = 1 \ \forall ij \in A$

Special cases

Transportation problem/Transshipment distribution of goods, warehouses-costumers

$|V_1| \neq |V_2|, \qquad u_{ij} = \infty$ for all $ij \in A$

$$\min \sum c_{ij} x_{ij}$$
$$\sum_i x_{ij} \geq b_j \qquad \forall j$$
$$\sum_j x_{ij} \leq a_i \qquad \forall i$$
$$x_{ij} \geq 0$$

if $\sum a_i = \sum b_i$ then $\geq / \leq$ become $=$
if $\sum a_i > \sum b_i$ then add dummy tank nodes
if $\sum a_i < \sum b_i$ then infeasible

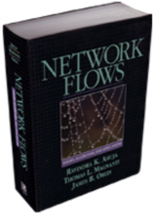
Multi-commodity flow problem ship several commodities using the same network, different origin destination pairs separate mass balance constraints, share capacity constraints, min overall flow

$$
\begin{aligned}
\min \textstyle\sum_k \mathbf{c}^k \mathbf{x}^k & \\
N\mathbf{x}^k \geq \mathbf{b}^k & \qquad \forall k \\
\textstyle\sum_k \mathbf{x}_{ij}^k \leq \mathbf{u}_{ij} & \qquad \forall ij \in A \\
0 \leq \mathbf{x}_{ij}^k \leq \mathbf{u}_{ij}^k &
\end{aligned}
$$

What is the structure of the matrix now? Is the matrix still TUM?

Application Example
Ship loading problem

Plenty of applications. See Ahuja Magnanti Orlin, Network Flows, 1993

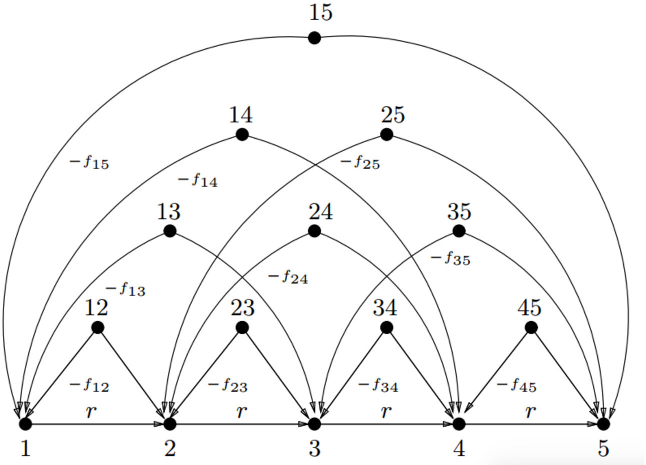


- A cargo company (eg, Maersk) uses a ship with a capacity to carry at most r units of cargo.
- The ship sails on a long route (say from Southampton to Alexandria) with several stops at ports in between.
- At these ports cargo may be unloaded and new cargo loaded.
- At each port there is an amount b_{ij} of cargo which is waiting to be shipped from port i to port $j > i$
- Let f_{ij} denote the income for the company from transporting one unit of cargo from port i to port j.
- The goal is to plan how much cargo to load at each port so as to maximize the total income while never exceeding ship's capacity.

Application Example: Modeling

- n number of stops including the starting port and the terminal port.
- $N = (V, A, \mathbf{l} \equiv \mathbf{0}, \mathbf{u}, \mathbf{c})$ be the network defined as follows:
  - $V = \{v_1, v_2, ..., v_n\} \cup \{v_{ij} : 1 \leq i < j \leq n\}$
  - $A = \{v_1 v_2, v_2 v_3, ... v_{n-1} v_n\} \cup \{v_{ij} v_i, v_{ij} v_j : 1 \leq i < j \leq n\}$
  - capacity: $u_{v_i v_{i+1}} = r$ for $i = 1, 2, ..., n-1$ and all other arcs have capacity ∞.
  - cost: $c_{v_{ij} v_i} = -f_{ij}$ for $1 \leq i < j \leq n$ and all other arcs have cost zero (including those of the form $v_{ij} v_j$)
  - balance vector: $b(v_{ij}) = b_{ij}$ for $1 \leq i < j \leq n$ and the balance vector of $b(v_i) = -b_{1i} - b_{2i} - ... - b_{i-1,i}$ for $i = 1, 2, ..., n$

Application Example: Modeling



Application Example: Modeling

Claim: the network models the ship loading problem.

- suppose that $t_{12}, t_{13}, \ldots, t_{1n}, t_{23}, \ldots, t_{n-1,n}$ are cargo numbers, where t_{ij} $(\leq b_{ij})$ is the amount of cargo the ship will transport from port i to port j and that the ship is never loaded above capacity.
- total income is

$$I = \sum_{1 \leq i < j \leq n} t_{ij} f_{ij}$$

- Let x be the flow in N defined as follows:
  - flow on an arc of the form $v_{ij}v_i$ is t_{ij}
  - flow on an arc of the form $v_{ij}v_j$ is $b_{ij} - t_{ij}$
  - flow on an arc of the form $v_i v_{i+1}$, $i = 1, 2, \ldots, n-1$, is the sum of those t_{ab} for which $a \leq i$ and $b \geq i + 1$.
- since t_{ij}, $1 \leq i < j \leq n$, are legal cargo numbers then x is feasible with respect to the balance vector and the capacity restriction.
- the cost of x is $-I$.

Application Example: Modeling

- Conversely, suppose that x is a feasible flow in N of cost J.
- we construct a feasible cargo assignment $s_{ij}, 1 \leq i < j \leq n$ as follows:
  - let s_{ij} be the value of x on the arc $v_{ij}v_i$.
- income $-J$

Outline

Shortest Path - Dual LP

$$z = \min \sum_{ij \in A} c_{ij} x_{ij}$$

$$\sum_{j:ji \in A} x_{ji} - \sum_{j:ij \in A} x_{ij} = 1 \qquad \textit{for } i = s \qquad (\pi_s)$$

$$\sum_{j:ji \in A} x_{ij} - \sum_{j:ij \in A} x_{ji} = 0 \qquad \forall i \in V \setminus \{s, t\} \qquad (\pi_i)$$

$$\sum_{j:ji \in A} x_{ji} - \sum_{j:ij \in A} x_{ij} = -1 \qquad \textit{for } i = t \qquad (\pi_t)$$

$$x_{ij} \geq 0 \qquad \forall ij \in A$$

Dual problem:

$$g^{LP} = \max \pi_s - \pi_t$$

$$\pi_j - \pi_i \leq c_{ij} \qquad \forall ij \in A$$

Hence, the shortest path can be found by potential values π_i on nodes such that $\pi_s = z, \pi_t = 0$ and $\pi_j - \pi_i \leq c_{ij}$ for $ij \in A$

Maximum (s,t)-Flow

Adding a backward arc from t to s:

$$
\begin{aligned}
z = \max x_{ts} & & \\
\sum_{j:ji\in A} x_{ij} - \sum_{j:ij\in A} x_{ji} = 0 & \qquad \forall i \in V & \qquad (\pi_i) \\
x_{ij} \leq u_{ij} & \qquad \forall ij \in A & \qquad (w_{ij}) \\
x_{ij} \geq 0 & \qquad \forall ij \in A &
\end{aligned}
$$

Dual problem:

$$
\begin{aligned}
g^{LP} = \min \sum_{ij\in A} u_{ij} w_{ij} & \\
\pi_i - \pi_j + w_{ij} \geq 0 & \qquad \forall ij \in A \\
\pi_t - \pi_s \geq 1 & \\
w_{ij} \geq 0 & \qquad \forall ij \in A
\end{aligned}
$$

$$
\begin{array}{r|cccccccc}
 & x_{e_1} & x_{e_2} & \cdots & x_{ij} & \cdots & x_{e_m} & & \\
 & c_{e_1} & c_{e_2} & \cdots & c_{ij} & \cdots & c_{e_m} & & \\
\hline
1 & -1 & . & \cdots & . & \cdots & . & = & b_1 \\
2 & . & . & \cdots & . & \cdots & . & = & b_2 \\
\vdots & \vdots & \ddots & & & & & = & \vdots \\
i & 1 & . & \cdots & -1 & \cdots & . & = & b_i \\
\vdots & \vdots & \ddots & & & & & = & \vdots \\
j & . & . & \cdots & 1 & \cdots & . & = & b_j \\
\vdots & \vdots & \ddots & & & & & = & \vdots \\
n & . & . & \cdots & . & \cdots & . & = & b_n \\
\hline
e_1 & 1 & & & & & & \leq & u_1 \\
e_2 & & 1 & & & & & \leq & u_2 \\
\vdots & \vdots & \ddots & & & & & \leq & \vdots \\
(i,j) & & & & 1 & & & \leq & u_{ij} \\
\vdots & \vdots & \ddots & & & & & \leq & \vdots \\
e_m & & & & & & 1 & \leq & u_m
\end{array}
$$

$$
\begin{aligned}
g^{LP} &= \min \sum_{ij \in A} u_{ij} w_{ij} && && (1)\\
\pi_i - \pi_j + w_{ij} &\geq 0 && \forall ij \in A && (2)\\
\pi_t - \pi_s &\geq 1 && && (3)\\
w_{ij} &\geq 0 && \forall ij \in A && (4)
\end{aligned}
$$

- Without (3) all potentials would go to 0.
- Keep w low because of objective function
- Keep all potentials low $\rightsquigarrow$ (3) $\pi_s = 0, \pi_t = 1$
- Cut C: on left =1 on right =0. Where is the transition?
- Vars w identify the cut $\rightsquigarrow$ $\pi_j - \pi_i + w_{ij} \geq 0 \rightsquigarrow w_{ij} = 1$

  $$
  w_{ij} = \begin{cases} 1 & if \;\; ij \in C \\ 0 & otherwise \end{cases}
  $$

  for those arcs that minimize the cut capacity $\sum_{ij \in A} u_{ij} w_{ij}$
- Complementary slackness: $w_{ij} = 1 \implies x_{ij} = u_{ij}$

Theorem

A strong dual to the max (st)-flow is the minimum (st)-cut problem:

$$\min_{X} \left\{ \sum_{ij \in A : i \in X, j \notin X} u_{ij} : s \in X \subset V \setminus \{t\} \right\}$$

Max Flow Algorithms

Optimality Condition

- Ford Fulkerson augmenting path algorithm $O(m|x^*|)$
- Edmonds-Karp algorithm (augment by shortest path) in $O(nm^2)$
- Dinic algorithm in layered networks $O(n^2m)$
- Karzanov's push relabel $O(n^2m)$

Min Cost Flow - Dual LP

$$\min \sum_{ij \in A} c_{ij} x_{ij}$$

$$\sum_{j:ji \in A} x_{ij} - \sum_{j:ij \in A} x_{ji} = b_i \qquad \forall i \in V \qquad (\pi_i)$$

$$x_{ij} \leq u_{ij} \qquad \forall ij \in A \qquad (w_{ij})$$

$$x_{ij} \geq 0 \qquad \forall ij \in A$$

Dual problem:

$$\max \sum_{i \in V} b_i \pi_i - \sum_{ij \in E} u_{ij} w_{ij} \tag{1}$$

$$-c_{ij} - \pi_i + \pi_j \leq w_{ij} \qquad \forall ij \in E \tag{2}$$

$$w_{ij} \geq 0 \qquad \forall ij \in A \tag{3}$$

- define reduced costs $\bar{c}_{ij} = c_{ij} + \pi_j - \pi_i$, hence (2) becomes $-\bar{c}_{ij} \leq w_{ij}$
- $u_e = \infty$ then $w_e = 0$ (from obj. func) and $\bar{c}_{ij} \geq 0$ (optimality condition)
- $u_e < \infty$ then $w_e \geq 0$ and $w_e \geq -\bar{c}_{ij}$ then $w_e = \max\{0, -\bar{c}_{ij}\}$, hence w_e is determined by others and irrelevant
- Complementary slackness th. for optimal solutions:
  each primal variable $\times$ the corresponding dual slack must be equal 0, ie, $x_e(\bar{c}_e + w_e) = 0$;
  - $x_e > 0$ then $-\bar{c}_e = w_e = \max\{0, -\bar{c}_e\}$,
    $x_e > 0 \implies -\bar{c}_e \geq 0$ or equivalently (by negation) $\bar{c}_e > 0 \implies x_e = 0$

  each dual variable $\times$ the corresponding primal slack must be equal 0, ie, $w_e(x_e - u_e) = 0$;
  - $w_e > 0$ then $x_e = u_e$
    $-\bar{c} > 0 \implies x_e = u_e$ or equivalently $\bar{c} < 0 \implies x_e = u_e$

  Hence:
  $\bar{c}_e > 0$ then $x_e = 0$
  $\bar{c}_e < 0$ then $x_e = u_e \neq \infty$

Min Cost Flow Algorithms

Theorem (Optimality conditions)

Let $\mathbf{x}$ be feasible flow in $N(V, A, \mathbf{l}, \mathbf{u}, \mathbf{b})$ then $\mathbf{x}$ is min cost flow in N iff $N(\mathbf{x})$ contains no directed cycle of negative cost.

- Cycle canceling algorithm with Bellman Ford Moore for negative cycles $O(nm^2UC)$, $U = \max|u_e|$, $C = \max|c_e|$
- Build up algorithms $O(n^2mM)$, $M = \max|b(v)|$

Outline

Assignment Problem

Input: a set of persons $P_1, P_2, \ldots, P_n$, a set of jobs $J_1, J_2, \ldots, J_n$ and an $n \times n$ matrix $M = [M_{ij}]$ whose entries are non-negative integers. Here M_{ij} is a measure for the skill of person P_i in performing job J_j (the lower the number the better P_i performs job J_j).
Goal is to find an assignment π of persons to jobs so that each person gets exactly one job and the sum $\sum_{i=1}^{n} M_{i\pi(i)}$ is minimized.

Matching Algorithms

Matching: $M \subseteq E$ of pairwise non adjacent edges

- bipartite graphs
- arbitrary graphs

- cardinality (max or perfect)
- weighted

Assignment problem $\equiv$ min weighted perfect bipartite matching $\equiv$ special case of min cost flow

bipartite cardinality

Theorem

The cardinality of a max matching in a bipartite graph equals the value of a maximum (s,t)-flow in N_{st}.

$\rightsquigarrow$ Dinic $O(\sqrt{n}m)$

Theorem (Optimality condition (Berge))

A matching M in a graph G is a maximum matching iff G contains no M-augmenting path.

$\rightsquigarrow$ augmenting path $O(\min(|U|,|V|),m)$

bipartite weighted
build up algorithm $O(n^3)$
bipartite weighted: Hungarian method $O(n^3)$

minimum weight perfect matching
Edmonds $O(n^3)$

Theorem (Hall's (marriage) theorem)

A bipartite graph $B = (X, Y, E)$ *has a matching covering* X *iff:*

$$|N(U)| \geq |U| \quad \forall U \subseteq X$$

Theorem (König, Egeavary theorem)

Let $B = (X, Y, E)$ *be a bipartite graph. Let* M^* *be the maximum matching and* V^* *the minimum vertex cover:*

$$|M^*| = |V^*|$$

Transportation Problem

Given: a set of production plants $S_1, S_2, ..., S_m$ that produce a certain product to be shipped to a set of re-tailers $T_1, T_2, ..., T_n$. For each pair (Si, Tj) there is a real-valued cost c_{ij} of transporting one unit of the product from S_i to T_j. Each plant produces $a_i, i = 1, 2, ..., m$, units per time unit and each retailer needs $b_j, j = 1, 2, ..., n$, units of the product per time unit.

Goal: find a transportation schedule for the whole production (i.e., how many units to send from S_i to T_j for $i = 1, 2, ..., m$, $j = 1, 2, ..., n$) in order to minimize the total transportation cost.

We assume that $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Summary

1. (Minimum Cost) Network Flows

2. Duality in Network Flow Problems

3. Assignment and Transportation